MULTIMEDIA UNIVERSITY

STUDENT ID NO

VENUE : _____

SEAT NO : _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2022/2023 (TERM ID 2210)

DCP 5201 – OBJECT ORIENTED PROGRAMMING

(All sections / Groups)

15 FEB 2023
9.00 a.m. – 11.00 a.m.
(2 Hours)

Question	Mark
1	
2	
3	
Total	

INSTRUCTIONS TO STUDENT

1. This question paper consists of 9 printed pages (including cover page).

2. Attempt all questions. The distribution of the marks for each question is given.

3. Print all your answers **CLEARLY** in the specific answer box provided for each question. Submit this question paper at the end of the examination.

QUESTION 1 [10 marks]

Q1(a). Implement mySwap() which swaps the values of two integer variables.

Sample Run 1:
```
a = 22     b = 77
a = 77     b = 22
```

[5]

```
#include <iostream>
using namespace std;

// Function prototype
// Write your code from here



int main()
{
  int a = 22;
  int b = 77;

  cout << "a = " << a << "  ";
  cout << "b = " << b << endl;

  mySwap( a, b );

  cout << "a = " << a << "  ";
  cout << "b = " << b << endl;

  return 0;
}

// Function definition
// Write your code from here
```

Q1(b). Write a program which uses a pointer variable m as a dynamically allocated integer array to store integers depending on the number of integers required by the user. The program will ask the user to enter the integers to be stored into the array and produce the output in the sample below.
Hint : Remember to free up the allocated memory before the program ends.

Sample Run 1:

Continued...

```
Enter the number of integers the array has to store : 5
Enter integer 1 : 2
Enter integer 2 : 3
Enter integer 3 : 4
Enter integer 4 : 5
Enter integer 5 : 6
Echoing the array
2 3 4 5 6
```

[5]

```
#include <iostream>
using namespace std;

int main( )
{

    int d1;
    cout << "Enter the number of integers the array has to store : ";
    cin >> d1;

    int* m;









    return 0;
}
```

Continued...

QUESTION 2 [10 marks]

Q2(a) Implement the class PersonalInfo and the function displayPersonalInfo so that the program displays the following output.

Program output:

```
Name: NoName
Address: NoAddress
Age: 0
Phone: NoPhone

Name: Ah Kau
Address: 48 Friendly Street
Age: 50
Phone: 555-8484

Name: Muthu
Address: 191 Apple Mountain Road
Age: 30
Phone: 555-2222
```

[6]

```
#include <iostream>
#include <string>
using namespace std;

class PersonalInfo
{
   private:
      string name;
      string address;
      int age;
      string phone;

   public:
      PersonalInfo()
      {
        // Write your code here









      }

      PersonalInfo(string n, string addr, int a, string p)
      {
         // Write your code here









      }
```

Continued...

```
      void setName(string n)
      { name = n; }

      void setAddress(string a)
      { address = a; }

      void setAge(int a)
      { age = a; }

      void setPhone(string p)
      { phone = p; }

      string getName() const
      { return name; }

      string getAddress() const
      { return address; }

      int getAge() const
      { return age; }

      string getPhone() const
      { return phone; }
};

void displayPersonalInfo( PersonalInfo obj )
{
   // Write your code here









}

int main()
{
   PersonalInfo me;
   PersonalInfo friend1("Ah Kau", "48 Friendly Street",
                        50, "555-8484");
   PersonalInfo friend2("Muthu", "191 Apple Mountain Road",
                         30, "555-2222");

   displayPersonalInfo( me );
   displayPersonalInfo( friend1 );
   displayPersonalInfo( friend2 );

   return 0;
}
```

Q2(b). What is the output of the following code?

Continued...

```
#include<iostream>
using namespace std;

class A
{
   public:
       A() { cout << "A is created" << endl; }
};
class B : public A
{
   public:
       B() { cout << "B is created" << endl; }
};

class C : protected B
{
   public:
       C() { cout << "C is created" << endl; }
};
class D : private C
{
   public:
       D() { cout << "D is created" << endl; }
};

int main()
{
    D test;

    return 0;
}
```

[2]

Q2(c). Referring to the code below and without declaring any new objects or variables, complete the program given. The program produces the output as shown in the sample run using the pointer variable x.

Sample Run 1:
```
This is B.
```

[2]

Continued...

```
#include <iostream>
using namespace std;

class A
{
   public:
      virtual void display();
};

class B: public A
{
   public:
      virtual void display();
};

void A::display()
{
   cout << "This is A." << endl;
}

void B::display()
{
   cout << "This is B." << endl;
}

int main()
{
   A* x;



   return 0;
}
```

Continued...

QUESTION 3 [10 marks]

Q3(a). Complete the program below which compares two objects of class `Amount` using an overloaded `==` operator and displays the results.

Sample Run 1:
```
2. Equal
```

[4]

```
#include <iostream>
using namespace std;

class Amount
{
   private:
      string sign;
      int value;

   public:
        Amount()
        {
           sign = "";
           value = 0;
        }

        Amount( string s, int val )
        {
           sign = s;
           value = val;
        }

        // Write your code from here









};

bool operator==(Amount a, Amount b)
{
   // Write your code from here









}

int main()
{
    Amount money1 ("RM", 500 );
```

Continued...

```
    Amount money2 ("S$", 500 );
    Amount money3 ("S$", 500 );
    Amount money4;

    if( money1 == money2 ) { cout << "1. Equal" << endl; }
    if( money2 == money3 ) { cout << "2. Equal" << endl; }
    if( money3 == money4 ) { cout << "3. Equal" << endl; }
    return 0;
}
```

Q3(b). Complete the following program so that its output is "MMU FCI" in reverse.
Hint : If s is a string "abc", s.length() returns 3;

Sample Run 1:
ICF UMM

[6]

```
#include <iostream>
using namespace std;

class MString
{
   private:
      string data;

   public:

   MString()
   {
      data = "";
   }

   // Write your code from here









};

int main()
{

  MString s( "MMU FCI" );
  s.PrintReverse();

  return 0;
}
```

End of Page